GROUP NEAREST NEIGHBOR QUERIES AGAINST DATA PIRACY

K.PADMAPRIYA, DR.S.SRIDHAR

Abstract - Avoiding data piracy is the biggest challenge in the internet world of obtaining and sharing data about individuals. Spatial queries give an additional threat about locations of these queries because using these data, hackers can easily expose the confidential information about the querier. For example, four friends want to meet in a restaurant nearer to all their residence but they don't want to share their residence area to anyone else. A group nearest neighbour(GNN) query returns the result that minimizes the aggregate distance from all the users. But the challenge is to provide security for the data of their locations. in this paper we proposed a framework to present a solution where all the users can select a common location without sharing their original location. This can be achieved by getting the details about their choices of meeting place like a restaurant where they can all meet. We have discussed about novel method to avoid data piracy and experimented the results showing its effectiveness and efficiency is better than the state-of-the-art algorithms.

Keywords: Group Nearest Neighbor, Data Piracy, Garbled Circuit, Boolean Circuit.

Introduction: The development of positioning technolgies (eg. GPS) in mobile environment facilitates "the killer application"in location based services(LBS). on the other hand similar to online shopping or web searches, queries related with location dependent data will lead to retrieval of personal details like their financial position, their health conditions and their political influences etc. For example they want to search for a restaurant with bar facility without communicating their location to one another. This can be achieved in client-server model of well known anonymity measures such as believing a third-party anonymizer[1, 2],or k-anonymizer[3,4] or making use of cryptographic protocols. In case of believing third party anonymizer, there is a change of data leakage if it is not a really trusted server. In hand held devices with advanced technologies such as IEEE 802.11 or Bluetooth will increase this problem. To get a optimal solution for this problem, we proposed GNN with cryptographic methodology. Here they can easily find out the restaurant with bar facility without trusting the third party and to share the location dependent data to each other. Geometrically, GNN answers for this query with with a set of data (p1, p2, ... pn} retrieve the set of places(q1,q2...,qm} with minimum aggrrgate distance from each point to each person. Using Euclidean distance we can compute it as s1 = $\sum_{i=1}^{n} d$(pi, q1), s2 = $\sum_{i=1}^{n} d$(pi, q2), ... , sk = $\sum_{i=1}^{n} d$(pi, qk). Then we get the minimum aggregate value of these functions using min(f1,f2,fk). It signifies the location of group of users. In this paper, we proposed the methodology to keep away from data piracy and evaluated its performance.

Related Work: We have splitted our existing approaches into location based services and cryptographical methods.

a. **Location based services(LBS) -** Many researchers studied [8,12,19]several mehods to secure data against piracy. Mostly they are based on cantralized server where third-party trusted server over there to act as a privacy guard for the users. Since it is storing all the informatins in a single place, it is very easy to hack the data. Hence some of the decentralized architectures [7,11,14] remove the role of third trusted party and secure their data with the help of its peers. Most of these approaches [7, 11, 10, 14,16] uses P2P network applying puzzled user's location within a circle or rectangle in addition to the location of the users 'request. So that the original location of the user is protected[7,10,11]. However the disadvantage of this method is that the users should have trust on their peers. But all these approaches either centralized or decentralized are done only for the single user but not for the group of users utilizing LBS.

b. **Cryptographical methods -** In the absence of third trusted party, the researchers tried to compute the function(x1,x2, ... ,xn) for individual bits x1, x2, .. , xn without revealing its original bits to anyone else. This has been named as secure function evaluation which provides communication among more than one party. They[19, 20, 21] have further divided this into two categories – semi-honest model and dis-honest model. In semi-honest model two/multi parties involved in this computation will obey the protocol absolutely but it tried to acquire the transcripts of their communication with other parties. In dis-honest model a part of the parties involved in this computation will not obey the protocol. Many protocols and solutions are developed in this approach mainly based on Yao's protocol[19] or the Goldreich, Micali and Wigderson(GMW)[20] protocol. Our paper make use of Yao's protocol developed for semi-honest model.

Proposed work: We developed a framework for securing the data against piracy in GNN computation especially in LBS which does not have third trusted party and it is based on P2P with commonly untrusted peers.

We divided our work into two parts – the centralized method and the distributed method.

We have conducted experiments on it to evaluate its performance which results into reasonable cost and easy to apply in real life.

Our methodology: We make use of Garbled circuits from Yao[19] to provide solution for secure function evaluation problem. Also we used some notations from[21,22] based upon garbled circuits.

Garbled circuits – assume that there are m parties where each party consists of n bit input. Consider the Boolean circuit function returns one n bit value. Assume the function has been represented by garbled Boolean circuit G consists of 2-gates for input and outputs a single bit value.

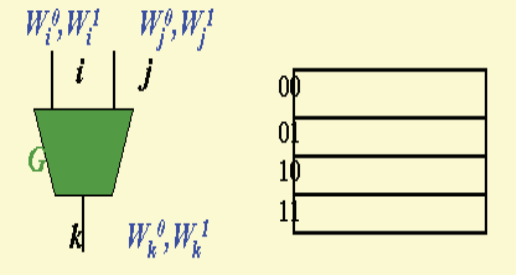

Fig. 1. Garbled circuit

In figure 1. the total number of wires are denoted as W where wires o, ... , mn-1 denotes the input whereas wires W-m, ..., W-1 denotes the output. Consider the input bit α before getting puzzled is denoted as b$\alpha \in \{0,1\}$. Each wire consists of two signals correlated with it. Signals are produced randomly but numbered with the same value of wire number. It has standard symmetric encryption keys size. Hence if α is said to be a wire then the correlated signals will be s2 α and s2 $\alpha + 1$. The value of each signal's plain text is associated with its semantic variable which can be denoted by ω. This semantic variable can be selected randomly but it should be kept as confidential. For more clear, consider a wire α, and its semantic variable $\omega\alpha \in \{0, 1\}$ can be selected randomly then $\omega\alpha$ be the semantics of s2 α and $\overline{\omega\alpha}$ be the semantics of s2 α+1.

The input wires send the signals as garbled inputs related with its corresponding actual input bits. For a wire α, the garbled inputs should be in the form of $\delta\alpha = s2\alpha + b\alpha \oplus \omega\alpha$and $\delta\alpha^{\prime} = s2\alpha + b\alpha \oplus \overline{\omega\alpha}$. The circuit is computed with four gate labels for each and every gate which has been generated by supplying the signals correlated with the wire to the pseudo random creator and calculate the XOR of the generated strings. The labels can be treated as truth table for gates with 2 input wires. Let C as the pseudo random creator which takes k bit inputs and outputs k+2mk bit string. Consider X, Y and Z are the first k, next mk and last mk of C's outputs respectively. Then xj=X(sj), yj=Y(sj), zj=Z(sj) where $0 \leq j \leq$W-1. The logic function XOR can be represented as $\oplus$. $\odot$ denotes the other logic functions such as AND, OR etc. If the incoming wires of a gate i are α and β for left and right respectively and γ is the outgoing wire, then it should be a=2α, b=2β and c=2γ. We can form the gate labels as

$$Ai=ga\oplus gb\oplus\begin{cases} sc & if\ \omega\alpha\odot\omega\beta=\omega\gamma \\ sc+1 & otherwise \end{cases}$$

$$Bi=ha\oplus gb+1\oplus\begin{cases} sc & if\ \omega\alpha\odot\overline{\omega\beta}=\omega\gamma \\ sc+1 & otherwise \end{cases}$$

$$Ci=ga+1\oplus hb\oplus\begin{cases} sc & if\ \overline{\omega\alpha}\odot\omega\beta=\omega\gamma \\ sc+1 & otherwise \end{cases}$$

$$Di=ha+1\oplus hb+1\oplus\begin{cases} sc & if\ \overline{\omega\alpha}\odot\overline{\omega\beta}=\omega\gamma \\ sc+1 & otherwise \end{cases}$$

The circuit evaluator considers all the gate labels mentioned above and the garbled inputs from all the users. If the evaluator has two input signals sa+p and ab+q for left and right incoming wires respectively where p,q ϵ {0,1}, then the computation will be

ga+p⊕gb+q⊕Ai	if p=0 and q=0
ha+p⊕gb+q⊕Bi	if p=0 and q=1
ga+p⊕hb+q⊕Ci	if p=1 and q=0
ha+p⊕hb+q⊕Di	if p=1 and q=1

Our aim is to learn the garbled circuits to answer GNN queries in semi-honest method in two ways – distributed and centralized.

1. **Distributed way of garbled circuits** – the whole region is partitioned into number of groups of locations and users. Two users are said to be responsible for a group of locations in creating and calculating the Boolean circuit. Suppose there are l locations and n users, the users will be grouped as pair and hence n/2 groups have been formed. In the same way the locations l are also having n/2 groups. When all the users are vigorously participating in creation and calculation of circuit, we have 2l/n locations for each group. This value will be varied according the number of

locations and number of users participating in each case

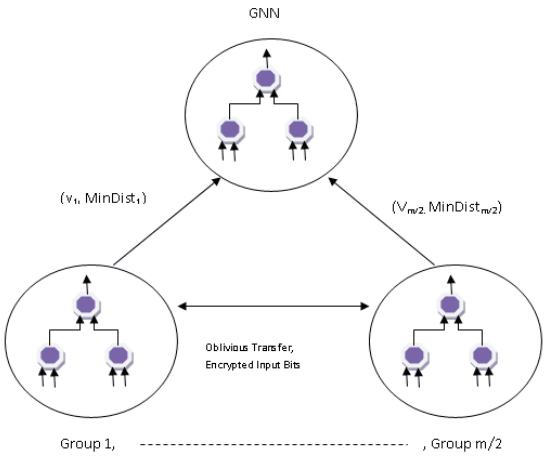

Fig. 2. Distributed approach for m/2 users

If the number of users is lesser than or equal to number of locations, we will get n/2 location groups and each user in user group is effectively contributing in creating and calculating the circuit. If the number of users is greater than the number of locations, then we will get the subset

of users who are responsible in circuit creation and calculation. Other users just having interaction with the subset of users to get input bits. In both the situations, a pair of users namely a and b ,will handle each group of locations. The encrypted circuit has been created by a and calculation has been done by b. Along with Oblivious transfer(OT), all users have to get their input bits and transmit it to b. At the same time a has to transmit the encrypted input bit to b. It shoud be carried out simultaneously till all group of locations are getting involved in it. Hence parallelism is achieved along with creation and calculation of circuits which consumes time. Figure 2 depicts it.

2. **Centralized way of garbled circuits –** the Boolean garbled circuit has been created by a single user a to represent the GNN function. The calculation has been done by an another user namely b. The rest of the users are getting encrypted input bits from a thru'oblivious transfer(OT) and send it to b. After getting encrypted inputs from all the users including a, b will start to calculate the circuit. Then after calculation, b will announce the output produced by the last gate of the Boolean circuit which has been treated as GNN. Hence either a or b don't know about the other users'input bits apart from their own bits. It has been achieved by OT. Figure 3 depicts this process for n users at the time.

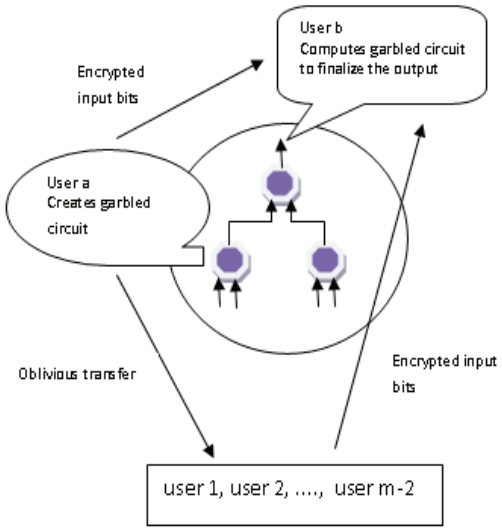

Fig. 3. Centralized approach with m/2 users

We developed a high performance GNN algorithm to answer the secured multi party function using distributed approach. Since the centralized way is the basic version, we have chosen distributed way to answer our GNN.

Steps involved in Distributed_Multiparty_GNN

1. Initialize the group of users, group of locations and successive pair of users.
2. Create the Boolean circuit
3. Compute the circuit to announce the final output bit.

In the first step, the number of users n and the number of location l are partitioned into groups and a successive pair of users is chosen who are responsible for creating the circuit(user a) and computing the circuit(user b) for each group of locations. If there are P users, then there are P/2 pairs to create and compute the circuit for GNN of all users in their group of locations. If n is lesser than or equal to l, then there are n/2 user groups. In the same way the locations l are also having n/2 groups. When all the users are vigorously participating in creation and calculation of circuit, we have 2l/n locations for each group. If n is greater than l, the locations are partitioned by l mod n locations in each group. Now we can start this stage by framing users group, locations groups and the successive pair of users(a and b) to create and compute the circuit.

In the second step, the Boolean circuit is created by each group with wires W. The user a creates signals and also semantics for all the wires and garbled encrypted input bits along with gate labels for each gate in the circuit. . Along with OT, all users have to get their encrypted input bits and transmit it to b. At the same time a has to transmit the encrypted input bit to b. It should be carried out simultaneously to achieve parallelism till all group of locations are getting involved in it.

In the third step, the evaluator b will compute the circuit of each group to finalize the output of that circuit. Whenever the evaluation of subcircuits are done, it has to be grouped together to build a final circuit to find out the location which has minimum distance from all the users.

Experimental Evaluation : Our framework has been developed in Java and implemented both distributed and centralized of garbled Boolean circuits for creation and computation. The number of locations has been varied between 200 to 2000 with the assumption of 10 users. Our experimental results show the scalability of our method. The garbled circuits we framed for GNN consists of 2 gates, one for a set of adder gates and another one for a comparator gate. Assume the length of input bit is 128 bits(n=128). The sum of distances to each location can be calculatedby s1=$\sum_{i=1}^{n} d(Pi, L1)$, s2=$\sum_{i=1}^{n} d(Pi, L2)$, ... , sl=$\sum_{i=1}^{n} d(Pi, Ll)$. The adder gates for each location have been created, hence we got ln bit adder gates i.e. one for each location. If we have 128 input bits, then we have to connect 128 1-bit adders in order. Hence After computing the sum of distances, we have to find out the minimum distance by comparing it with each other. It can be achieved by our n bit comparator gate. Hence finally we framed (l+1)n bit gates in total.

Figure 4 shows the time taken by our methodologies while creating the Boolean circuit. Figure 5 shows the the time taken by our methodologies while computing the circuit. The time taken during creation of circuit is high when compared with computing the circuit. This is because of involving many steps while creating the circuit such as creating random key and encrypting the original input bit. Also in distributed model, the number of users and locations are partitioned into groups, so that each

group involves creating and computing the circuit for their belonging group. Hence the work load is becoming higher than in case of centralized approach.

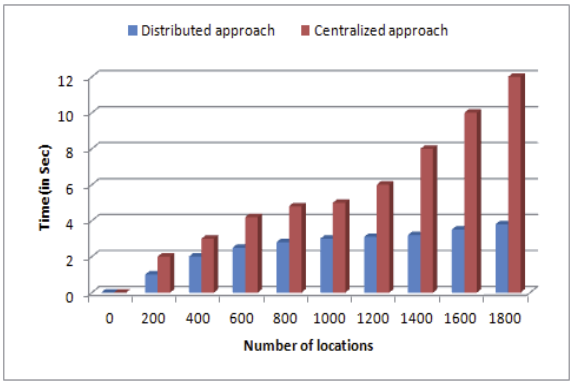

Fig. 4. Time taken during creation of Boolean circuit

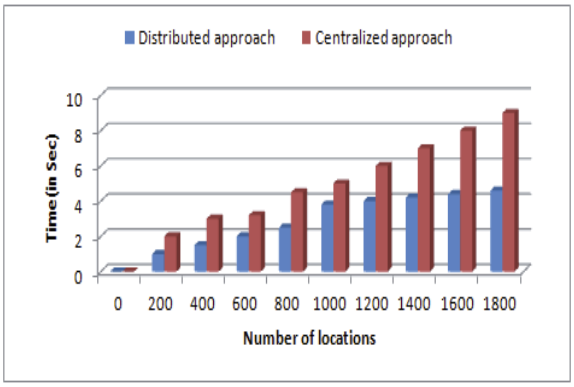

Fig. 5. Time taken during computation of Boolean circuit

While considering about the costs, both approaches are controlled by OT. Since the number of OTs is same in both the cases, there is no difference. Therefore the communication cost remains same for both distributed and centralized approaches.

Conclusion: In this paper, we have addressed the problem of data piracy in location based services and we discussed about providing the security for group nearest neighbours. Without providing all information about their residence to everyone how the group of people can find out the meeting place has been discussed. In previous wotk they had third trusted party to send and receive the data among themselves. Since the third trusted party is storing all data in a single place, it is highly risk to belive the third party. In our work, we achieved our goal without having any trusted third party. We used P2P model to secure data of all peers who are all involved in it. Our experimental results show the effectiveness and efficiency of our model and it outperforms well when compared with the existing methodologies.

References:

1. C. Y. Chow, M. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based services," in *Proc. of the ACM GIS*, 2006, pp. 247–256.
2. G. Ghinita, P. Kalnis, and S. Skiadopoulos, "PRIVE: Anonymous location-based queries in distributed mobile systems," in *Proc. of the 1st Int. Conference on World Wide Web (WWW)*, 2007, pp. 371–380.
3. B. Gedik and L. Liu, "Privacy in mobile systems: A personalized anonymization model," in *Proc. of ICDCS*, 2005, pp. 620–629.
4. M. Mokbel, C. Chow, and W. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proc. of VLDB*, 2006, pp. 219–229.
5. Khoshgozaran and C. Shahabi, "Private information retrieval techniques for enabling location privacy in location-based services," in *Proc. of Privacy in Location-based Applications*, 2009, pp. 59–83.
6. M. Yiu, C. Jensen, X. Huang, and H. Lu, "Spacetwist: managing the trade-offs among location privacy, query performance and query accuracy in road networks," in *Proc. of ICDE '08*, 2008, pp. 366–375.
7. M. Yiu, N. Mamoulis, and D. Papadias, "Aggregate nearest neighbour queries in road networks," in *Proc. of IEEE TKDE*, 2005, pp. 820–833.
8. D. Papadias, Y. Tao, J. Zhang, and N. Mamoulis, "Query processing in spatial network databases," in *Proc. of VLDB*, 2003, pp. 802–813.
9. Yao, "How to generate and exchange secrets," in *Proc. of 27th IEEE Symposium on Foundations of Computer Science FOCS '86*, 1986, pp. 162–167.
10. O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proc. of 19th ACM Symposium on Theory of Computing*, 1987, pp. 218–229.
11. G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVÉ: Anonymous location-based queries in distributed mobile systems. In WWW, pages 371–389, 2007.
12. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In MobiSys, pages 31–42, 2003.
13. Guttman. R-trees: a dynamic index structure for spatial searching. In SIGMOD, pages 47–57, 1984.
14. T. Hashem and L. Kulik. Safeguarding location privacy in wireless ad-hoc networks. In Ubicomp, pages 372–390, 2007.
15. G. R. Hjaltason and H. Samet. Ranking in spatial

databases. In SSD, pages 83–95, 1995.
16. H. Hu and J. Xu. Non-exposure location anonymity. In ICDE, pages 1120–1131, 2009.
17. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In SSTD, pages 239–257, 2007.
18. H. Li, H. Lu, B. Huang, and Z. Huang. Two ellipse-based pruning methods for group nearest neighbor queries. In GIS, pages 192–199, 2005.
19. Yao, "How to generate and exchange secrets," in *Proc. of* 27*th IEEE Symposium on Foundations of Computer Science FOCS '86*, 1986, pp. 162–167.
20. O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proc. of 19th ACM Symposium on Theory of Computing*, 1987, pp. 218–229.
21. S. Namnandorj, H. Chen, K. Furuse, and N. Ohbo. Efficient bounds in finding aggregate nearest neighbors. In DEXA, pages 693–700, 2008.
22. D. Papadias, Q. Shen, Y. Tao, and K. Mouratidis. Group nearest neighbor queries. In ICDE, page 301, 2004.
23. D. Papadias, Y. Tao, K. Mouratidis, and C. K. Hui. Aggregate nearest neighbor queries in spatial databases. TODS, 30(2):529–576, 2005.
24. N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbour queries. In SIGMOD, pages 71–79, 1995.
25. M. Strassman and C. Collier. Case study: The development of the find friends application. In Location-Based Services, pages 27–40. 2004.
26. K. F. Yanmin Luo, Hanxiong Chen and N. Ohbo. Efficient methods in finding aggregate nearest neighbor by projection-based filtering. In ICCSA, pages 821–833, 2007.

***

K.Padmapriya/ Research Scholar/ Sathyabama University/
Dr.S.Sridhar/Professor & Dean – CCCF/R.V.College of Engineering.